Nonparametric Methods

- Introduction
- Nonparametric Density Estimation
- Histogram Estimator
- Kernel Estimator
- K-Nearest Neighbor Estimator

Introduction

- In Bayesian classifier
  - If the underlying pdfs $P(C_i)$, $p(\mathbf{x}|C_i)$ are unknown
    - Need to be estimated from available training data $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$
- In regression
  - To approximate the unknown $f(\mathbf{x})$
    - $\mathbf{y} = f(\mathbf{x}) + \varepsilon, \varepsilon \sim N(0, \sigma^2)$ from $X = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$
- Estimation of $p(\mathbf{x}|C_i)$ or $p(y|\mathbf{x})$
  - 1) Parametric model
    - $p(\mathbf{x}|C_i) \equiv p(\mathbf{x}|C_i; \boldsymbol{\theta}_i)$ or $p(y|\mathbf{x}) \equiv p(y|\mathbf{x}, \boldsymbol{\theta})$
      - Ch4, Ch5
  - 2) Nonparametric estimation
    - Without the assumption about the form of the underlying densities
    - Estimating $p(\mathbf{x}|C_i)$, $p(C_i|\mathbf{x})$ or $p(y|\mathbf{x})$ directly

Nonparametric Density Estimation (1)

- Nonparametric methods
  - The density is not assumed to be characterized by parameters
  - Suppose $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ are drawn i.i.d. according to $p(\mathbf{x})$
  - Let P be the probability that $\mathbf{x}$ will fall in a certain region R is
    - $P = \int_R p(\mathbf{x}')d\mathbf{x}'$
      - P is a smoothed or averaged version of the density function $p(\mathbf{x})$
    - How do we estimate P?

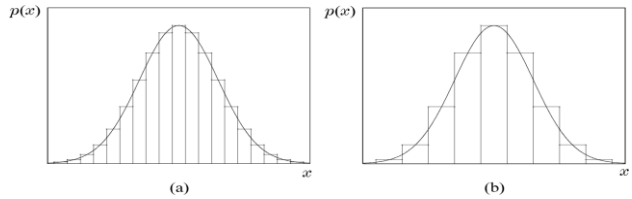


Fig. 2.18 [Theodoridis 09]

Nonparametric Density Estimation (2)

- Estimation of P
  - The probability of having k of N samples fall in R is given by the binomial law
    - $P_k = \binom{N}{k} P^k (1-P)^{N-k}$
    - The binomial distribution for k peaks very sharply about the mean
  - The expected value for k is
    - $E\{k\} = \sum_{k=0}^{N} kP_k$
      $= \sum_{k=0}^{N} k \binom{N}{k} P^k (1-P)^{N-k}$
      $= NP \sum_{k=1}^{N} \binom{N-1}{k-1} P^{k-1}(1-P)^{N-k} = NP$
    - We expect the ratio $\frac{k}{N}$ will be a very good estimate for P

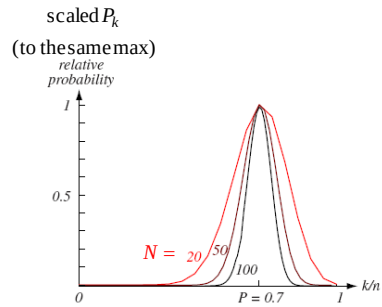


Fig. 4.1 [Duda 01]

Nonparametric Density Estimation (3)

- Estimation of P (cont.)
  - The probability P that $\mathbf{x}$ will fall in a certain region R is approximated by the frequency ratio
    - $P \approx \frac{k_N}{N}$
      - k_N: the number of samples located within R
  - This estimation is unbiased and asymptotically consistent
    - $E\left\{\frac{k_N}{N}\right\} = \frac{1}{N}E\{k_N\} = P$
    - $E\left\{\left(\frac{k_N}{N} - P\right)^2\right\} = E\left\{\left(\frac{k_N}{N}\right)^2\right\} - P^2 = \cdots = \frac{P^2}{N}\left(N - 1 + \frac{1}{P}\right) - P^2 = \frac{P(1-P)}{N}$

Nonparametric Density Estimation (4)

- Density estimation of $p(\mathbf{x})$
  - If $p(\mathbf{x})$ is continuous and nearly constant within the small region R
    - $P = \int_R p(\mathbf{x}')d\mathbf{x}' \approx p(\hat{\mathbf{x}})V$
      - where $V = \int_R d\mathbf{x}'$ and $\hat{\mathbf{x}}$ is a point within R
    - $p(\mathbf{x}) \equiv p(\hat{\mathbf{x}}) \approx \frac{1}{V}\frac{k_N}{N}$
  - 1-D case
    - The 1-D feature space is divided into successive bins of length h
      - $\hat{p}(x) = \hat{p}(\hat{x}) \approx \frac{1}{h}\frac{k_N}{N}$
        - » $|x - \hat{x}| \leq \frac{h}{2}$
    - As $N \to \infty, \hat{p}(x)$ converges to $p(x)$ when
      - $h_N \to 0$ (small bin $\Rightarrow p(x)$ is nearly constant in the bin)
      - $k_N \to \infty$ ($\because k_N \approx PN$)
      - $k_N/N \to 0$ (P must be finite)

Nonparametric Density Estimation (5)

- $p(\hat{\mathbf{x}}) \approx \frac{1}{V}\frac{k_N}{N}$
  - Kernel estimator (also called Parzen windows)
    - Fixing the volume V, and determining k_N from the data
  - KNN estimator
    - Fixing k_N, and determining V from the data

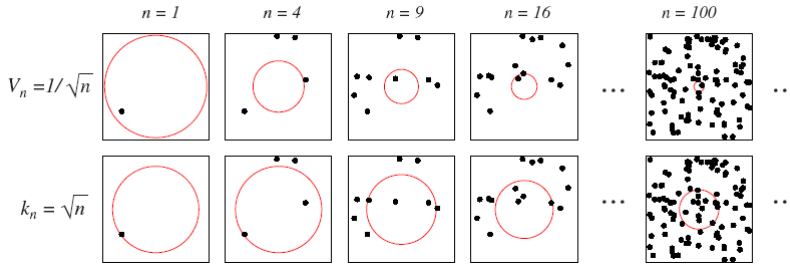


Fig. 4.2 [Duda 01]

Histogram Estimator (1)

- 1-D case $\hat{p}(x) \approx \frac{1}{h}\frac{k_N}{N}$
  - Once the histogram is computed, the data set can be discarded

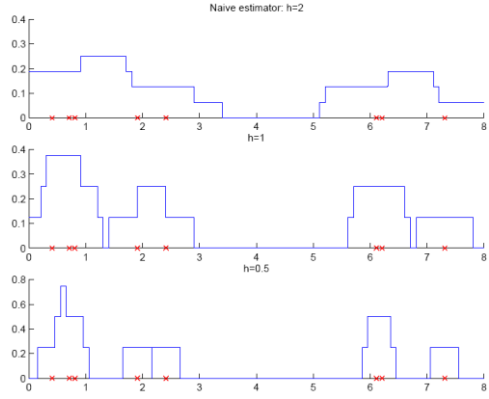


Fig 8.2 [Alpaydin, 2014]

Histogram Estimator (2)

- Limitations
  - The estimated density is discontinuous
    - Dependent on the width of bin
  - The approach is scaling with dimensionality
    - M bins in each of the l-dimensional space => M^l bins

The original data is drawn from a mixture of 2 Gaussians
- Small bin
  spiky density
- Large bin
  smooth density that fails to capture the bimodal property

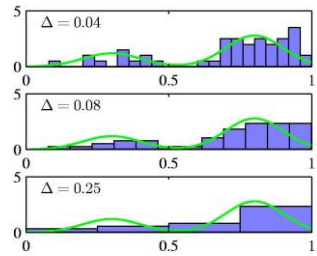


Fig. 2.24 [Bishop 06]

Kernel Estimator (1)

- The volume around $\mathbf{x}$ is considered a fixed l-D hypercube
  - Volume $V = h^l$
    - h: length of side of the hypercube
  - Window function
    - $\varphi(\mathbf{x}) = \begin{cases} 1, & \text{if } |x_j| < \frac{1}{2}, j = 1, \dots, l \\ 0, & \text{otherwise} \end{cases}$

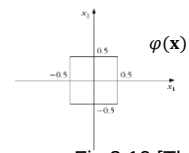
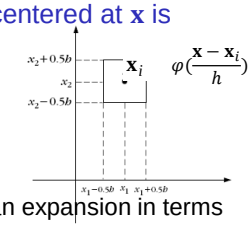


Fig 2.19 [Theodoridis 09]

  - The number of samples inside the hypercube centered at $\mathbf{x}$ is
    - $k_N(\mathbf{x}) = \sum_{i=0}^{N} \varphi(\frac{\mathbf{x}-\mathbf{x}_i}{h})$
    - Then
      - $\hat{p}(\mathbf{x}) = \frac{1}{V}\frac{k_N(\mathbf{x})}{N} = \frac{1}{h^l}\frac{1}{N}\sum_{i=0}^{N} \varphi(\frac{\mathbf{x}-\mathbf{x}_i}{h})$



    - To approximate a continuous function $p(\mathbf{x})$ via an expansion in terms of discontinuous window function

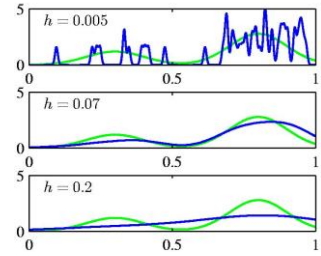
Kernel Estimator (2)

- To approximate a smoother density model
  - Using smooth functions with
    - $\varphi(\mathbf{x}) \geq 0$
    - $\int \varphi(\mathbf{x}) d\mathbf{x} = 1$
      - $\varphi(\mathbf{x})$: kernels, potential functions or Parzen windows
  - e.g.
    - Gaussian kernel $\varphi(\mathbf{x}) = N(0, \mathbf{I})$
    - The density model
      - $\hat{p}(\mathbf{x}) = \frac{1}{h^l}\frac{1}{N}\sum_{i=0}^{N}\varphi(\frac{\mathbf{x}-\mathbf{x}_i}{h})$

$$= \frac{1}{N}\sum_{i=0}^{N}\frac{1}{(2\pi)^{\frac{l}{2}}h^l}\exp\left(-\frac{(\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}-\mathbf{x}_i)}{2h^2}\right)$$

    - i.e. the unknown pdf is approximated as an average of N Gaussians, each one centered at a different point of the training set

Fig. 2.25 [Bishop 06]



Kernel Estimator (3)

- Kernel density estimator (KDE)
  - Or Parzen window density estimator
    - $\hat{p}(\mathbf{x}) = \frac{1}{h^l}\frac{1}{N}\sum_{i=0}^{N}\varphi(\frac{\mathbf{x}-\mathbf{x}_i}{h})$
    - No model fitting is required
      - The bandwidth can be tuned by cross-validation
    - But, takes a lot of memory and time

Fig. 14.16 [Murphy]:
Other smoothing kernels

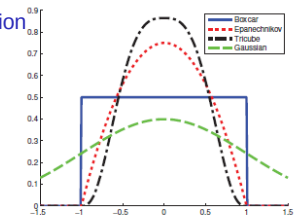


Figure 14.16 A comparison of some popular smoothing kernels. The boxcar kernel has compact support but is not smooth. The Epanechnikov kernel has compact support but is not differentiable at its boundary. The tri-cube has compact support and two continuous derivatives at the boundary of its support. The Gaussian is differentiable, but does not have compact support. Based on Figure 6.2 of (Hastie et al. 2009). Figure generated by `smoothingKernelPlot`.

Kernel Estimator (4)

- Choice of smoothing parameter
  - Small h
    - $\hat{p}(x)$ is the superposition of N sharp pulses centered at the samples
  - Large h
    - $\hat{p}(x)$ is the superposition of N broad, slowing changing functions
    - Structure in the probability density estimate is lost

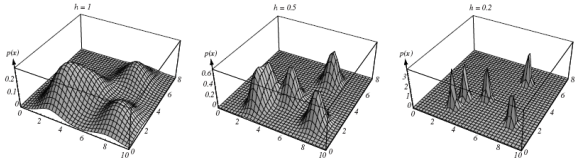


Fig 4.4 [Duda 01]

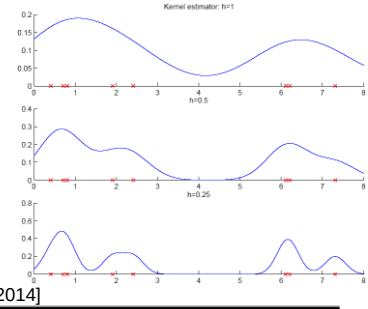


Fig 8.3 [Alpaydin, 2014]

Kernel Estimator (5)

- Convergence of the mean
  - $\lim_{N\to\infty} E\{\hat{p}(\mathbf{x})\} = \lim_{N\to\infty} \frac{1}{h^l}\frac{1}{N}\sum_{i=0}^{N} E\{\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)\}$

    $\xrightarrow{N\to\infty} \int_{\mathbf{x}'} \frac{1}{h^l}\varphi\left(\frac{\mathbf{x}-\mathbf{x}'}{h}\right) p(\mathbf{x}')d\mathbf{x}'$

    $= \int_{\mathbf{x}'} \delta_N(\mathbf{x}-\mathbf{x}')p(\mathbf{x}')\,d\mathbf{x}'$ Convolution
  - As $h \to 0$
    - $\frac{1}{h^l}\varphi\left(\frac{\mathbf{x}-\mathbf{x}'}{h}\right) \to \delta(\mathbf{x}-\mathbf{x}')$ delta function centered at $\mathbf{x}$
    - Because
      - The width of $\varphi\left(\frac{\mathbf{x}-\mathbf{x}'}{h}\right) \to 0$, $\frac{1}{h^l} \to \infty$, $\int_{\mathbf{x}'} \frac{1}{h^l}\varphi\left(\frac{\mathbf{x}-\mathbf{x}'}{h}\right)d\mathbf{x}' = \int_{\mathbf{u}} \varphi(\mathbf{u})d\mathbf{u} = 1$
    - $\hat{p}(\mathbf{x})$ is an unbiased estimate of $p(\mathbf{x})$ in the limiting case
      - $E\{\hat{p}(\mathbf{x})\} = \int_{\mathbf{x}'} \delta_N(\mathbf{x}-\mathbf{x}')p(\mathbf{x}')\,d\mathbf{x}' \to p(\mathbf{x})$

Kernel Estimator (6)

- Convergence of the variance
  - $E\{\hat{p}^2(\mathbf{x})\} - (E\{\hat{p}(\mathbf{x})\})^2 = \frac{1}{N^2}\sum_{i=0}^{N} E\left\{\frac{1}{h^{2l}}\varphi^2\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)\right\} - (E\{\hat{p}(\mathbf{x})\})^2$

    $= \frac{1}{Nh^l}\int_{\mathbf{x}'} \frac{1}{h^l}\varphi^2\left(\frac{\mathbf{x}-\mathbf{x}'}{h}\right) p(\mathbf{x}')d\mathbf{x}' - (E\{\hat{p}(\mathbf{x})\})^2$

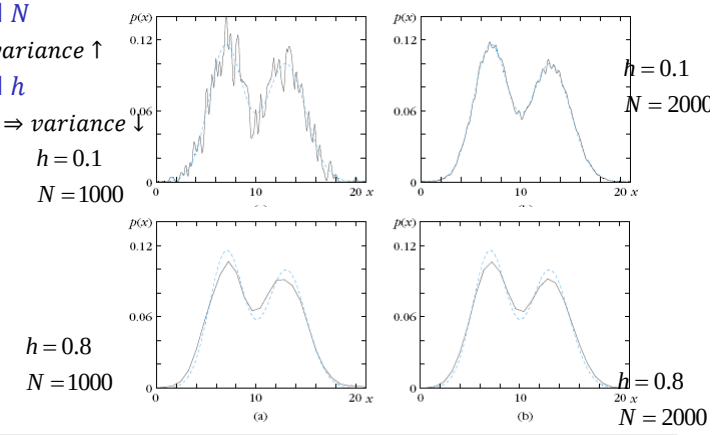
    $\leq \frac{sup(\varphi(\cdot))(E\{\hat{p}(\mathbf{x})\})}{Nh^l}$ (dropping the 2nd term)

  - The estimation is asymptotically consistent if
    - $h \to 0$ and $hN \to \infty$
  - For a fixed N
    - $h \downarrow \Rightarrow variance \uparrow$
  - For a fixed h
    - $N \to \infty \Rightarrow variance \downarrow$

Kernel Estimator (7)

- Example
  - $\varphi(x) \sim N(0,1)$
  - For a fixed N
    - $h \downarrow \Rightarrow variance \uparrow$
  - For a fixed h
    - $N \to \infty \Rightarrow variance \downarrow$

Fig. 2.20 & Fig. 2.21 [Theodoridis 09]



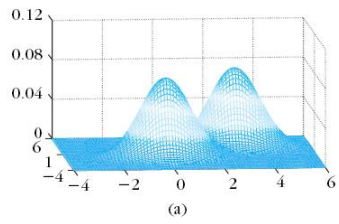
Kernel Estimator (8)

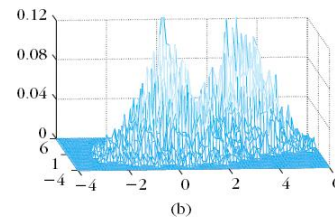
Fig. 2.22 [Theodoridis 09]

$\varphi(\mathbf{x}) \sim N(0, \mathbf{I})$

Ground truth

$p(\mathbf{x})$



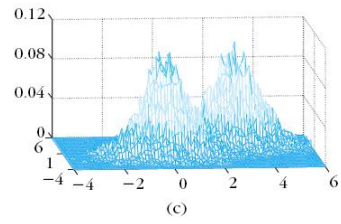


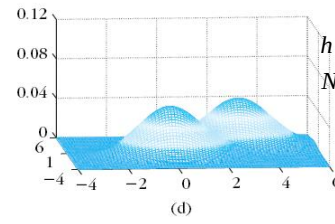
$h = 0.05$

$N = 1000$

$h = 0.05$

$N = 20000$





$h = 0.8$

$N = 20000$

Kernel Estimator (9)

- Nonparametric classification
  - No assumptions about the distributions
  - Given $\{X_1, X_2, \ldots, X_M\}$
  - Class-conditional density
    - $p(\mathbf{x}|C_j) \equiv \frac{1}{|X_j|} \sum_{\mathbf{x}_i \in X_j} \frac{1}{h^l} \varphi(\frac{\mathbf{x}-\mathbf{x}_i}{h})$
  - Classifier
    - $C' = \underset{C_j}{\operatorname{argmax}}\, p(\mathbf{x}|C_j) P(C_j)$

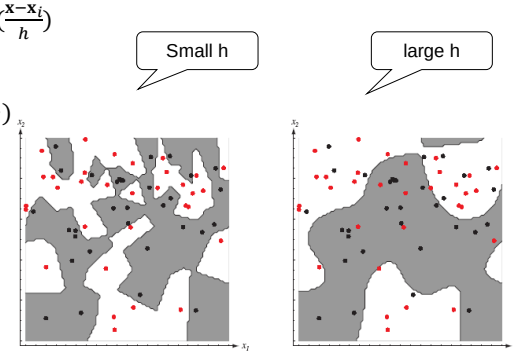


Fig. 4.8 [Duda 01]

KNN Estimator (1)

- KNN density estimation
  - The number of points k_N falling inside the volume is fixed
    - Let the cell about $\mathbf{x}$ grow until it captures k_N samples
    - $\hat{p}(\mathbf{x}) \approx \frac{1}{V(\mathbf{x})} \frac{k_N}{N}$
  - The size of volume $V(\mathbf{x})$ around $\mathbf{x}$ will be adjusted
    - In low-density area, the volume will be large
    - In high-density area, the volume will be small

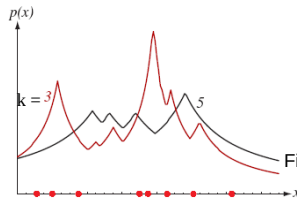
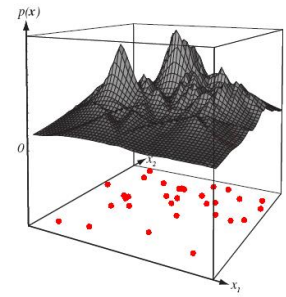


Fig. 4.10 [Duda 01]

Fig. 4.11 [Duda 01]



k = 5

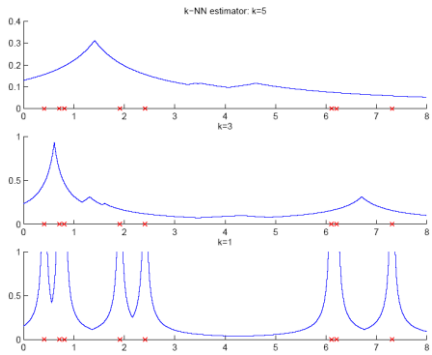


Fig 8.4 [Alpaydin, 2014]

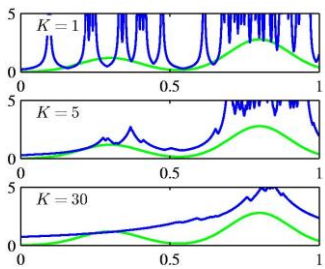
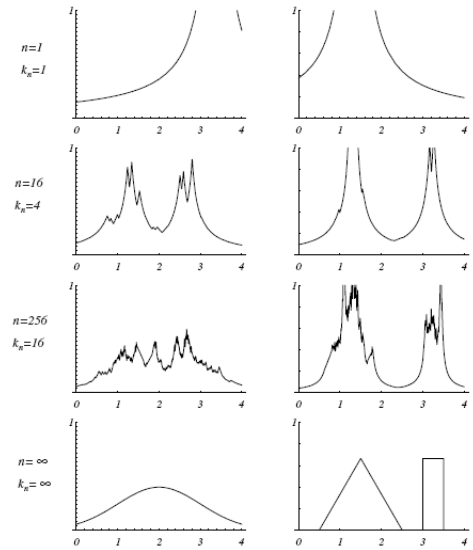


Fig. 2.26 [Bishop 06]



Gaussian Bimodal

Fig. 4.12 [Duda 01]

KNN Estimator (2)

- Example (p. 58 [Theodoridis 09])
  - Assume $P(C_1) = P(C_2)$
  - Let $k = 5$
  - Given $\mathbf{x} = \begin{bmatrix} 0.7 \\ 0.6 \end{bmatrix}$
    - $\hat{p}(\mathbf{x}|C_1) = \frac{1}{V_1(\mathbf{x})}\frac{k}{N_1} = \frac{1}{\pi(\sqrt{0.08})^2}\frac{5}{59}$
    - $\hat{p}(\mathbf{x}|C_2) = \frac{1}{V_2(\mathbf{x})}\frac{k}{N_2} = \frac{1}{\pi(\sqrt{0.02})^2}\frac{5}{61}$
    - $\hat{p}(\mathbf{x}|C_1) < \hat{p}(\mathbf{x}|C_2)$
    - $\hat{P}(C_1|\mathbf{x}) < \hat{P}(C_2|\mathbf{x})$
    - $\mathbf{x} \rightarrow C_2$

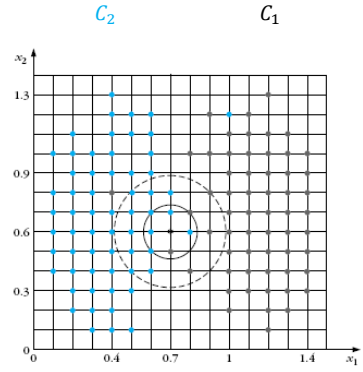


Fig. 2.24 [Theodoridis 09]

KNN Estimator (3)

- A variation of KNN density estimation
  - Suppose the cell of volume V around $\mathbf{x}$ captures k samples
    - If k_i of k samples are labeled C_i
      - $\sum_{j=1}^{M} k_j = k$
    - Let $\hat{p}(\mathbf{x}|C_i) = \frac{1}{V_i(\mathbf{x})}\frac{k_i}{N_i}$

The fraction of the samples within the cell that are labeled C_i

    - $\hat{P}(C_i|\mathbf{x}) = \frac{\hat{p}(\mathbf{x}|C_i)P(C_i)}{\sum_{j=1}^{M}\hat{p}(\mathbf{x}|C_j)P(C_j)} = \frac{\frac{1}{V_i(\mathbf{x})}\frac{k_i}{N_i}\frac{N_i}{N}}{\sum_{j=1}^{M}\hat{p}(\mathbf{x}|C_j)P(C_j)} = \frac{\frac{1}{V_i(\mathbf{x})}\frac{k_i}{N}}{\frac{1}{V_i(\mathbf{x})}\frac{k}{N}} = \frac{k_i}{k}$
  - The minimum-error-rate classifier
    - To assign $\mathbf{x}$ to the class that received the largest vote among the k nearest neighbors
      - $\mathbf{x} \rightarrow C_i$ if $\hat{P}(C_i|\mathbf{x}) \geq \hat{P}(C_j|\mathbf{x}), \forall j \neq i$
      - $\mathbf{x} \rightarrow C_i$ if $k_i \geq k_j, \forall j \neq i$

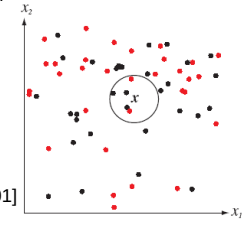


Fig. 4.15 [Duda 01]

KNN Estimator (4)

- Examples

Fig. 2.25 [Theodoridis 09]

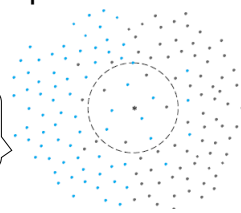


Fig. 2.27 [Bishop 06]

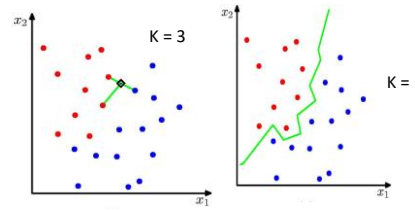
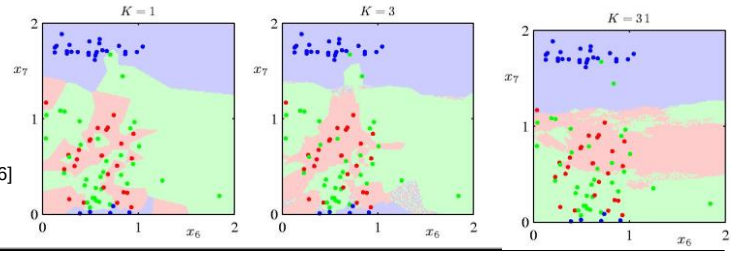


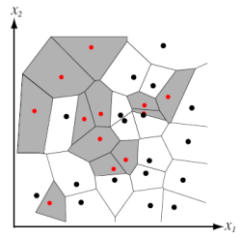
Fig. 2.28 [Bishop 06]

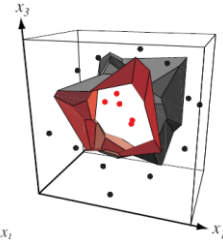


KNN Estimator (5)

- Nearest-neighbor rule (i.e. $k = 1$)
  - Given N labeled prototypes
    - Assign $\mathbf{x}$ to the class of its nearest neighbor
  - The training set defines a partition of the l-dimensional space into N regions
    - $R_i = \{\mathbf{x} | d(\mathbf{x}, \mathbf{x}_i) < d(\mathbf{x}, \mathbf{x}_j), j \neq i\}$
    - Voronoi tessellation

$l = 2$





$l = 3$

Fig. 4.13 [Duda 01]

KNN Estimator (6)

- Error rate for NN-rule
  - Is never more than twice the minimum error rate
    - As $N \to \infty$
      - $P_B(e) \le P_{NN}(e) \le P_B(e)\left(2 - \frac{M}{M-1}P_B(e)\right) \le 2P_B(e)$

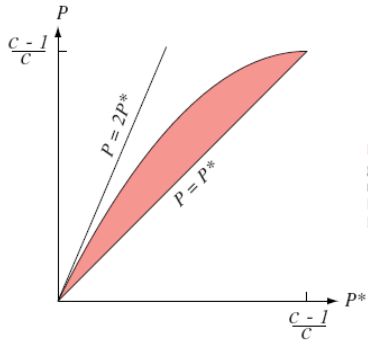


FIGURE 4.14. Bounds on the nearest-neighbor error rate P in a c-category problem given infinite training data, where P^* is the Bayes error (Eq. 52). At low error rates, the nearest-neighbor error rate is bounded above by twice the Bayes rate. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Fig. 4.14 [Duda 01]

KNN Estimator (7)

- Computational complexity of NN rule
  - Brute-force searching:
    - $O(lN^2)$
      - Each distance calculation is $O(l)$
      - Inspect each training sample and retain the nearest one
  - Partial distance
    - Calculate the distance using some subset r of the full l dimensions
      - Terminate a distance calculation to any prototype once its partial distance is greater than the full Euclidean distance to the current closest prototype
  - Search tree
    - Calculate the distance to one or a few “entry” prototypes
      - Consider only the prototypes which are selectively linked to the closest entry prototypes